Alejo M. Pacalso Memorial National High School Student Information System

Divine L. Aguilar[1], Dr. Ellen M. Halover[2],
and Hydi D. Toyeng[1]

[1] Faculty, School of Information and Technology
University of Baguio, Baguio City, Philippines
Email: divine@e.ubaguio.edu

[2] Head, Quality Assurance Office,
University of Baguio, Baguio City, Philippines

Abstract

Educational institutions regularly collect, process, and generate a considerable amount of varied data. When managed appropriately, data provide information useful for planning, decision-making, policy-making, research, and innovation. One of the most common tools used to manage data is the Information System (IS), an organized combination of people, hardware, software, communications networks, data resources, policies, and processes. Alejo M. Pacalso Memorial National High School (AMPMNHS), which uses a traditional way of keeping and managing records, recognized the need for an IS to address difficulties encountered by the staff and teachers in maintaining the school's growing data and in producing timely reports. In response, the School of Information Technology (SIT) developed the Student Information System (SIS) to handle AMPMNHS's student data and generate reports requested by students, alumni, parents, and reports needed by the school and required by the Department of Education (DepEd). The proponents used Rapid Application Development (RAD), which comprises four (4) phases: the Requirements Planning phase, User Design phase, Construction phase, and Cut-over phase. RAD allows the project sponsor and user representatives' active participation from start to end of the development. The SIS is a responsive web application. Its primary functions are 1) management of students' personal information, section, enrolled subjects, grades, transfer credentials for transfer students, the school's curriculums, subjects, class sections, employee information, and 2) timely generation of reports. The SIS was assessed using different testing techniques. It complies with the users' requirements and possesses the software qualities specified in ISO 25010:2011. The system is ready for use by AMPMNHS.

Keywords: Alejo M. Pacalso Memorial National High School (AMPMNHS), Student Information System, Rapid Application Development

Introduction

Educational institutions collect and generate a considerable amount of data relevant to students, employees, course schedules, courses, curriculums, curricular and co-curricular activities, performance evaluation, tuition, and payments regularly. Data are vital in any educational institution's operations; thus, it must be effectively secured and maintained. When managed appropriately, data provides information that is very useful for planning, decision-making, policy-making, research and innovation, and other educational/business activities. While some institutions use information systems to manage all or some of their data, some schools are still stuck with the traditional way of collecting, storing, and processing data. When requests are made, for example, a permanent record, the staff in charge locates the student record from the shelves, encodes it in the computer, and prints the report. The old way of managing data still works but requires too much time and effort to complete every task. Also, printed records may fade or be destroyed/damaged during disasters such as fire or flood and may not be retrievable. If there are no other copies of the printed document in separate locations, the school may face many problems.

An information system (IS) is an organized combination of people, hardware, software, communications networks, data resources, and policies and procedures that stores, retrieves, transforms, and disseminates information in an organization (O'Brien & Marakas, 2011). Information Systems are utilized by organizations to support business processes and operations, decision making, and competitive advantage strategies. IS is used in sales and marketing, operations, research and development, finance, human resource, productivity, information management, business intelligence, and other business aspects across industries. Ticketing systems, booking/reservations systems, accounting systems, enrolment systems, e-commerce systems, human resource information systems, payroll systems are examples of an IS.

One of the Department of Education's (DepEd) thrusts is to modernize education management and governance by automating core systems and processes (Dino, 2017). DepEd's move is in line with the literature review findings on impacts of management information system on school administration conducted by Shah (2013), indicating a very positive impact of ICT use in educational management. MIS can supply administrators and teachers with the information required for informed planning, policy-making, and evaluation; MIS have change school management in the areas of leadership, decision making, workload, human resource management, communication, responsibility, and planning; and these systems can aid the school manager in determining the goals of the school, constructing strategic plans, distributing resources, and evaluating staff performance as well as organizational success (Shah, 2013).

DepEd maintains a website where advisories, announcements, DepEd Orders, memorandum, reports, and other communications are published. Some of these DepEd documents may be downloaded by any interested individual. In 2015, DepEd launched the Learner Information System (LIS) and Enhanced Basic Education Information System (EBEIS) and released DO 26, s. 2015 to guide the schools, teachers, and other users of the system. The systems aim to establish accurate and reliable learners and schools' registries to ensure data availability and information needed for planning and budgeting, allocating resources, and setting operational targets (DepEd, n.d.). Both LIS and EBEIS are being used today by DepEd and basic education schools. Other information systems needed by primary education schools in their operations are their responsibility.

Alejo M. Pacalso Memorial National High School (AMPMNHS), an adopted school of the University of Baguio, located in Bua, Tuding, Itogon, Benguet, offers junior and senior high school programs. The school keeps student records on shelves and retrieves them every time there is a need to extract data and generate a report. This traditional data management scheme often delays report generation; it is tedious and needs physical space. Since the school has computers and communications devices, and with the government's advocacy to use Information and Communications Technology (ICT), AMPMNHS thought of utilizing an information system to manage student data. The school's aim to utilize IS was articulated to the School of Information Technology (SIT), which became the basis for this study.

Significance of the Study

The study is aligned to the Department of Education's (DepEd) goal to modernize education management and governance, specifically its objective to automate core systems and processes.

The study addresses the need of Alejo M. Pacalso Memorial National High School (AMPMNHS) for a computerized system that will allow concerned staff members and teachers to encode, store, and manage student information such as personal information and grades, and to generate relevant reports such as Student Permanent Record for both junior and senior high school students. Having a computerized system for managing student records will lessen the staff's time and effort to generate reports requested by clients (former and current students or their parents/guardians; other schools; DepEd, higher education institutions). Consequently, staff can focus on other tasks and be more productive. The project aimed to help AMPMNHS in terms of providing fast and efficient service to its clients. Other high schools having similar processes may also use and benefit from the application.

The project is beneficial to the School of Information Technology and the university because it extended its expertise to an adopted school. It addressed the need of AMPMNHS.

The project had been a learning experience for the proponents because they obtained firsthand experience in the different software development phases. While doing the project, they honed their personal and technical skills in requirements elicitation and specification, database design, coding and testing, actual deployment of the software, and maintenance. The improvements in the proponent's skills and their experiences shall cascade to the School of Information Technology and its students.

Objectives of the Study

The objective of the study is to develop an application that will facilitate the management of student data and the generation of reports at the Alejo M. Pacalso Memorial National High School. To achieve this, the team shall:

- Determine the functional requirements of the system
- Develop an appropriate database design
- Construct and test the system

Methodology

The study falls under applied research, where existing knowledge is used to develop a new product. The proponents observed software design and development standards and principles while developing the AMPMNHS Student Information System.

Rapid Application Development (RAD) was used in the design and development of the system. In RAD, the project sponsor and representatives of the target users participate actively from start to end. RAD comprises four phases: the Requirements Planning phase, User Design phase, Construction phase, and Cut-over phase. With RAD, the proponents and the school's representatives maintained continuous communication to ensure the correctness of the functional requirements, appropriateness of the user interfaces, and accuracy of generated data and information.

The Requirements Planning phase includes system planning and systems analysis phases of the Systems Development Life Cycle (SDLC). During this phase, initial interview sessions with the school representatives were conducted to determine the school's objectives regarding the system to be developed. After that, the team members and the school representatives discussed and agreed on the project scope, constraints, timeline, and system requirements.

During the User Design phase, the team ensured interaction with some of the target users, including the staff members and a representative of the teachers, to determine their specific needs, how they manage student information and what reports they produce. After that, a prototype representing the system's different processes, including the inputs and outputs, was created and presented to the users to obtain their feedback, suggestions, and approval. Acquisition of actual forms and a face-to-face interview with the users were the primary techniques to elicit system requirements. Actual student and employee data were not requested as they are not necessary to develop the system. Only details (purpose, formats (e.g., numeric, letters, alphanumeric), sample, length, etc.) of the data were asked from the school representatives.

The Construction Phase focuses on application development, including integration of modules and testing. The developers used PHP, Visual Studio Code, and MySQL during the development. In the initial stage of construction, the developers requested the active participation of the target users. They were asked to use the completed modules and give comments and recommendations as they do a system walkthrough. However, when the quarantine started due to the COVID-19, the users' system walkthrough was not possible. Instead, the developers built the system based on previous communications, forms, and templates at hand.

The Cutover Phase is the implementation phase, where the system is deployed in the target users' environment. This phase also includes the orientation of the different users. In this phase, errors and other user issues may still be encountered by the users. The SIS has been thoroughly tested, but due to the restrictions that have been implemented by the government and the risks to both the developers and the school representatives, the system is yet to be deployed.

Sample / Population of the Study

The RAD methodology requires the active involvement of target users in the different phases of development. In the Student Information System developed for the AMPMNHS, two (2) administrative staff members and the teacher-in-charge of ICT were the primary contacts and data and information providers. Teachers who were present during meetings also joined the discussions and walkthroughs.

Data Gathering Tools

The primary techniques used to gather relevant information were interview and document analysis. The interview guide questions revolved around what the users need and how they want the system to behave. Relevant forms such as Student Information Sheet, reports like Form 137, and written guidelines were provided to the developers.

The developers used a checklist to determine if the Student Information System possesses the characteristics of a quality software based on ISO 25010:2011 Systems and Software Engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models.

Data Gathering Procedures

The proponents set meetings with the school representative(s) to gather requirements and other relevant information through the SIT Outreach Coordinator. Most of the meetings were held in AMPMNHS, and a few meetings were conducted in the School of Information Technology, University of Baguio. Meetings were scheduled and organized based on the availability of both the school representative(s) and the developers. Initially, the meetings' objectives were to gather data and clarify processes and entries in the forms/templates. As the development progressed, meetings were conducted to present prototypes and testing of completed modules. In the testing phase, the developers accomplished the Software Quality checklist.

Treatment of Data

Thematic analysis was used to process and analyze gathered data from interviews and documents. The developers familiarized themselves with the different datasets and terminologies used by the school. Data were clustered according to nine (9) themes, namely, Users of the System, Reports, Student Personal Information, Transfer Data, Curriculum and Subjects, Sections, Employee Information, Grades, and Others. In the end, the developers discarded all data under Others because they were not relevant to the first eight (8) themes or were related but beyond the project's scope. The data under each theme were reviewed and further grouped into smaller clusters. These data were used in generating user requirements which were transformed into functional requirements.

Data from the Software Quality Checklist were processed using simple counting and by computing the percentage of compliance. A percentage below 100 was interpreted as Partial Compliance, and Full Compliance if the percentage was 100.

Ethical Considerations

All gathered data/information was obtained upon the approval and with the assistance of the principal's representative(s). The proponents treated all gathered data and those generated during the software development with strict confidentiality and used them solely to develop the system. The proponents strictly followed the school's data processing procedures and the provisions of the Data Privacy Act of 2012. Security measures were integrated into the system to safeguard the data in the database. The system was designed to ensure data reliability, meaning only authorized school personnel can access, add and alter certain or all data. One security measure implemented in the system was the need for an authorized higher authority to confirm the change/deletion before executing the process.

The system documentation and the final research manuscript will be given to the school for their reference. The study's results will be disseminated to the UB community (with the school's permission) through the UB publication and research presentation organized by the Research and Development Center.

Results and Discussion

The Student Information System (SIS) developed for Alejo M. Pacalso Memorial National High School (AMPMNHS) is a web application that can be accessed/opened from any device with a web browser. The application allows authorized staff members and teachers to encode, store, and manage student data and information such as personal information, section, subjects, and grades, and generate relevant reports like Student Permanent Record for both Junior and Senior High School students and Form 137 of students and alumni. The requirements gathering and analysis phase resulted in the functional requirements presented in Table 1. Functional requirements are software features or functions that the developers must implement. These are customized according to the processes of the organization or how the organization wants to do things.

Table 1

Functional requirements

Functional Requirements	**Description**
Login / Logout	"Login" allows a user access to the system. This is a security measure designed to prevent unauthorized access to confidential data. "Logout" signs out the user and redirects to the login page.
The functional requirements below apply to the following entities: Student, Employee, Curriculum, Subject, School Year.	
Show records	"Show records" displays active records in a tabular format. The last column of the table contains link buttons of actions (View, Archive, Delete) that can be performed by the user.
Search	"Search" allows a user to look for a record or group of records using a search key. If the search is successful, all records that contain the search key will be displayed on the grid; otherwise, no record will be displayed.
Add new record	"Add new record" displays a page where the user enters the required data for a particular record.
View a record	"View a record" displays a page that contains the details of a particular record.
Edit a record	"Edit a record" allows the user to alter editable data in a record. It involves two actions, namely, changing the current data and saving the changes made. For readily editable data, the user may change and save. For restricted data, permission from an authorized user is required before saving the changes. The system logs changes in student records.
Confirm student record update	"Confirm record update" is triggered when a user attempts to save changes made to restricted data. The page shows a summary of changes made (Original data, edited data) and will require the name, designation (automatically shown), and password of the authorized school personnel who will confirm the changes.
Employee	
Archive an employee	"Archive employee" allows the user to move employee records out of the active list. Archived data is stored so that at any time, it can be brought back into service.
Curriculum & Subject	
Delete a curriculum / subject	Only a curriculum/subject that has not been used/linked to other records can be deleted.
Add a subject to the curriculum	"Add a subject in the curriculum" facilitates adding of a subject to a particular curriculum. This functionality is used when the school adopts a new curriculum.

Sections	
Add section	"Add section" allows a user to create a new section for a particular school year or semester. Each section is populated by a set of automatically pre-selected subjects based on the chosen curriculum and grade level. The pre-selected subjects can be modified such that unnecessary subjects/inactive subjects can be removed.
Filter Section	"Filter section" allows a user to filter the displayed sections by SY, Grade, Semester; default current school year.
Edit Section	"Edit Section" allows a user to modify a section's details (remedial status, section name, and adviser).
Add subject inside the section.	"Add subject inside the section" allows a user to insert a subject into a certain section.
Student Section Assignment and Grades Module	
Assign a student to a section	"Assign student to section" allows a user to link a student to a particular section. This function is synonymous with enrolling a student in a block section.
Download Grade Sheet	"Download Class Record" allows the user to download the grade sheet. This will serve as a template for uploading the grades into the system.
Upload Grade	"Upload Grade" allows the user to upload the grade sheet into the system.
Reports	
Preview report	"Preview Report" displays data based on a set filter(s) or search criterion/criteria. The report contains the school's letterhead, the report title, the date and time the report was generated, and the user who generated it based on the username used during login.
Print report	"Print report" is triggered by clicking the Print button on the Preview Report page. The previewed report will be printed using the default or selected printer.
Export report	"Export report" is an alternative to Print. This allows the displayed data to be exported as MS Word or pdf files.

Software Design. Software design is a two-step process: 1) Architectural design (also referred to as top-level design and high-level design) describes how software is organized into components, and 2) Detailed design describes the desired behavior of these components (Bourque & Fairley, 2014). The AMPMNHS Student Information System, a web database application, is built around the three-tier architecture model shown in Figure 1.

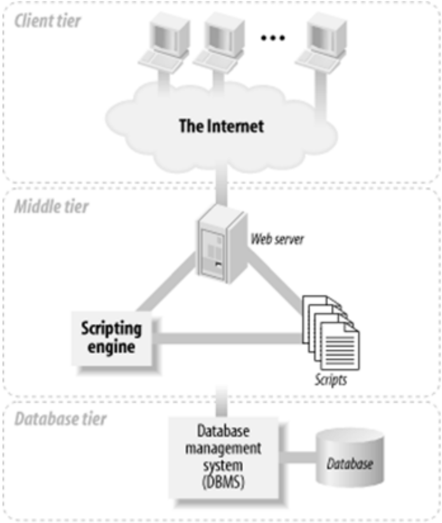


Figure 1. AMPMNHS architecture model

Three-tier software architecture organizes applications into three (3) logical and physical computing tiers, namely the presentation tier, where the user interacts with the system; the application tier, where data processing happens; and the data tier, where the data associated with the application is stored and managed (IBM Cloud Education, 2020). Oracle (n.d.) asserts two (2) advantages of three-tier architecture: high performance and a high degree of flexibility in deployment platform and configuration. Moreover, three-tier architecture offers faster development, improved scalability, improved reliability, and improved security (IBM Cloud Education (2020).

At the base of the system is the database tier, consisting of the database management system (DBMS) used to create, maintain, and provide controlled access to the database. The system database is stored in a database server and managed using MySQL, an open-source version of Structured Query Language (SQL) that runs on Linux, UNIX, Windows, and Mac OS X operating systems.

The middle tier, which is built on top of the database tier, contains most of the application's logic. It manages the structure and content of the data displayed to the user, processes input from the user, and transforms it into queries on the database, adds state management to the Hypertext Transfer Protocol (HTTP), and integrates the Web with the database server. Three-tier applications have good reliability because of the middle tier, which performs all security-related tasks (Sharma, Kumar, & Agarwal, 2015). The middle-tier components are the web server, web scripting language, and the scripting language engine. The Student Information System uses Apache HTTP Server as its web server, Personal Hypertext Pre-processor (PHP) as its scripting language, and PHP Zend as its scripting engine. Apache is an open-source web server that is lightweight, fast, and scalable. It can handle simultaneous requests from user agents and is designed to run under multitasking operating systems. PHP is an open-source server-side scripting language embedded into static Hypertext Mark-up Language (HTML) that makes client tier integration easy. PHP Zend Scripting Engine is responsible for executing the compiled script or code, and it returns the output to the webserver interface.

The client tier presents data to and collects data from the user through web browsers such as Microsoft's Edge, Mozilla's Firefox, Apple's Safari, Google's Chrome, and Opera. The system can be accessed through the school's Intranet; when a user (browser) requests a page, it will be sent through the network to the Web server, then in return, the webserver gives the requested page, which will be displayed through the user's browser. If the request requires data to be obtained from the database, the Web server creates a query and forwards it to the database server, which processes the query and returns the results set when it is run against the database. Similarly, data entered at the user's station is passed through and stored in the database by sending it to the Web server, which passes it on to the database server, which commits the data to the database.

Database Design. A database comprises an organized collection of data (Bourque & Fairley, 2014). Designing a database is an equally important phase in the software development lifecycle because it affects the performance and maintainability of the system and the accuracy and integrity of data and information. Data modeling, in database design, is the process of constructing a data model for the data to be saved in the database.

The SIS database schema consists of twelve (12) tables. Normalization was applied during the process of determining tables and the columns (fields) and primary keys for each table. Microsoft (2020) defines normalization as the process of organizing data in a database, including creating tables and initiating relationships between those tables according to the rules designed both to safeguard the data and to make the database more flexible/adjustable by eliminating conflicting dependency and redundancy.

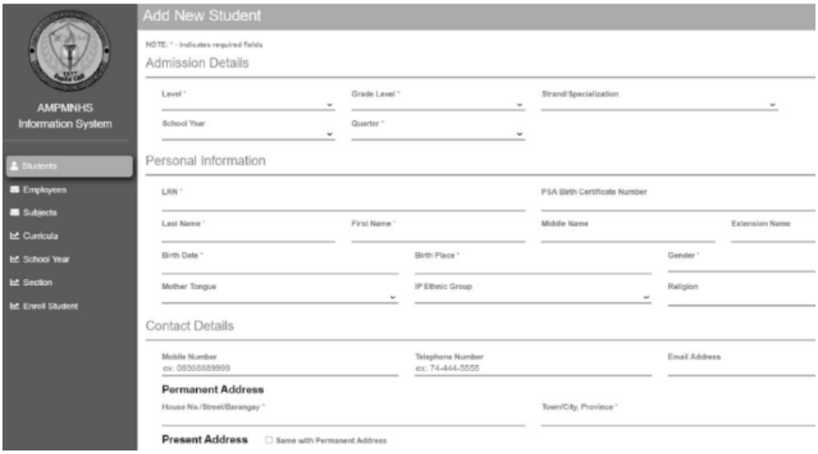
Table 2 is provided to have a better understanding of the tables and the database.

Table 2
The Database

Table Name	Description
Students	This table contains the personal information and admission data of students.
School_years	This table is used to record a school year. A new school year is added annually, and it becomes the default school year for all transactions in the system. It also contains the start and end of the school year.
Levels	Table Level contains the grade levels, classified as Junior High School or Senior High School, and the different SHS strands and specializations.
TranfereeGrades	This table stores students' transfer data such as previous school, grade level with its corresponding subjects, and grades.
Student_section_subjects	This records the subjects taken by a student based on his/her section.
Grades	This table stores a student's grades every quarter based on the subjects taken by the student in the Student_section_subject table.
Curricula	The Curricula Table contains information about a curriculum such as a curriculum code, level, strand and specialization, effectivity, and status.

Subjects	Subjects table is used to store a subject's code, description, and type.
Curricula_subjects	This table contains the subjects included in a curriculum.
Sections	The Sections Table stores the sections created for an upcoming or current school year and term. It also contains the assigned adviser per section.
Section_subjects	This table contains the subjects included in a section. The Section_subjects table is linked to the Student_section_subjects table to determine the student's section.
Employees	Employees Table contains some personal information of teaching and non-teaching employees who will be using the system.

User Interface Design. User Interface (UI) design is an indispensable part of the software design process. It ensures effective interaction between the users and the system. An actual UI of the AMPMNHS Student Information System, as seen in Figure 3, conforms to standards and general principles for a better user design and experiences like learnability, user familiarity, consistency, minimal surprise, recoverability, and user guidance. According to Bourque (2014), the UI design is "learnable" if the software is easy to learn and the users can rapidly start using the software with ease; employs "user familiarity" if the interface has utilized concepts and terms drawn from the experiences of the intended users of the software; is "consistent" if comparable operations are activated in the same way; ensures "minimal surprise" if the behavior of the software does not surprise users, and it does not implement unexpected behaviors; is "recoverable" if the software provides mechanisms allowing users to recover from errors; reflects "User Guidance" if the interface gives meaningful feedback when errors occur and provide context-related help to users.



NOTE: * - indicates required fields

Figure 3. Add New Student user interface

The AMPMNHS Student Information System's UI was designed to match the experience, skills, and expectations of the staff and teachers. In the study of Abugbah, Sanzogni & Poropat (2009), user characteristics are identified as one of the most important factors that affect user performance directly and indirectly within the IS environment. Interviews with the users to determine their general characteristics revealed that the staff members are quite proficient in using productivity tools and other applications that they are using in the office. On the other hand, the teachers use productivity tools for creating course modules, test questionnaires, and presentations. Since government agencies have been using online systems, the staff and most teachers have sufficient experience in navigating web-based systems. These user attributes were taken into consideration during the UI design phase. Moreover, the school's logo is reflected in the main pages of the system and on the reports.

System Construction and Testing. Software construction refers to the comprehensive creation of a working software through a combination of coding, verification, unit testing, integration testing, and debugging (Bourque

& Fairley, 2014). Figure 4 shows the sequence of the development of the different modules.

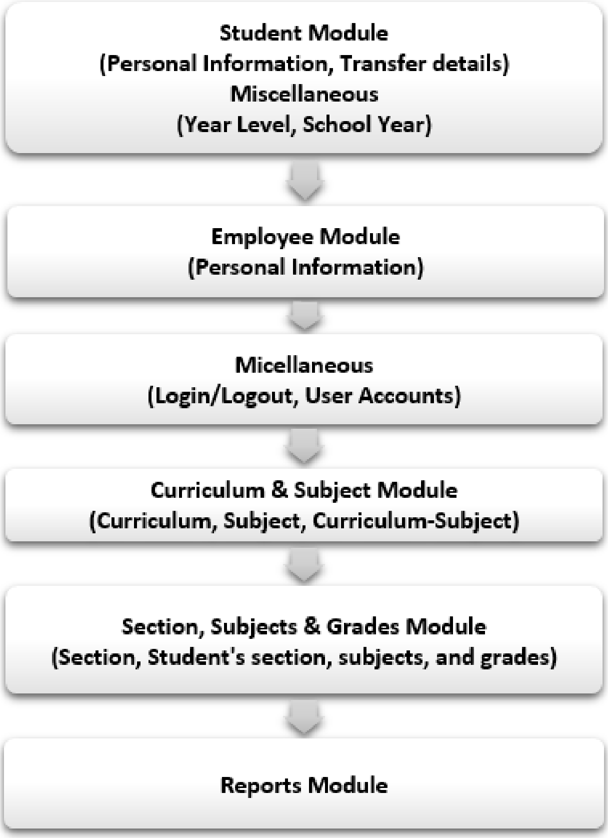

Student Module
(Personal Information, Transfer details)
Miscellaneous
(Year Level, School Year)
Employee Module
(Personal Information)
Micellaneous
(Login/Logout, User Accounts)
Curriculum & Subject Module
(Curriculum, Subject, Curriculum-Subject)
Section, Subjects & Grades Module
(Section, Student's section, subjects, and grades)
Reports Module

Figure 4. Development sequence

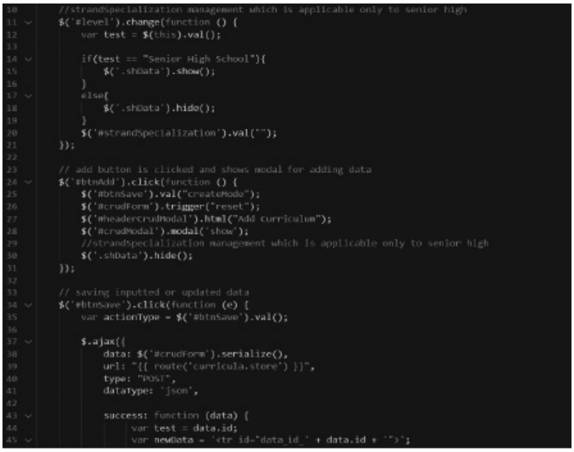
Some miscellaneous processes such as management of school year and grade level, and user accounts, login, and logout were coded with the Student Module and after the Employees Module, respectively, because the Student Module needs them and so that the software can be deployed and accessed by authorized staff/teachers even if the system is not complete. One of the challenging modules was the Section, Subjects & Grades Module because different scenarios had to be captured. In between developments were team meetings and consultations with the school representatives.

The developers adhered to software construction fundamentals specified by Bourque & Fairley (2014) in the Guide to the Software Engineering Body of Knowledge Version 3, namely, minimizing complexity, anticipating change, constructing for verification, reuse, and use of standards in construction.

The techniques used include the following:

- Simple and readable coding such that other developers can understand if the system will be handed over to them.
- Standard formats for documentation, naming convention, layout, and source code layout were consistently observed during coding.
- Handling of anticipated and exception errors
- Source code organization that facilitates construction for reuse

The application of the techniques mentioned earlier is reflected in Figure 5, a screenshot from the system's actual source code.



```
//strandSpecialization management which is applicable only to senior high
$('#level').change(function () {
    var test = $(this).val();

    if(test == "Senior High School"){
        $('.shData').show();
    }
    else{
        $('.shData').hide();
    }
    $('#strandSpecialization').val("");
});

// add button is clicked and shows modal for adding data
$('#btnAdd').click(function () {
    $('#btnSave').val("createMode");
    $('#crudForm').trigger("reset");
    $('#headerCrudModal').html("Add Curriculum");
    $('#crudModal').modal('show');
    //strandSpecialization management which is applicable only to senior high
    $('.shData').hide();
});

// saving inputted or updated data
$('#btnSave').click(function (e) {
    var actionType = $('#btnSave').val();

    $.ajax({
        data: $('#crudForm').serialize(),
        url: "{{ route('curricula.store') }}",
        type: "POST",
        dataType: 'json',

        success: function (data) {
            var test = data.id;
            var newData = '<tr id="data_id_' + data.id + '">';
```

Figure 5. Actual source code

Software testing involves testing the program on selected inputs to check it provides expected behaviors on a finite/limited set of test cases, suitably selected from the usually infinite execution domain (Bourque & Fairley, 2014). In the construction phase, unit and integration testing were done to address faults, failures, and errors at the early development stage. Unit testing is the verification of the functioning of a particular testable unit in the software independent of other parts. Link (2004) states that early and continuous unit testing is crucial for high-quality software and low defect rates. Integration testing is the process of verifying the operation of two or more components or subsystems of software. The developers used the incremental integration approach, where testing is done after every subsystem is added until all the system components have been adjoined. Since the system is quite small, manual unit and integration testing were performed. After integrating all the system components, testing ensued in the development environment. The developers tested the system's functionality using different scenarios. The testing yielded minor issues such as data entry errors that were not captured, missing error messages / ambiguous error messages, and some design issues. The developers fixed the defects and retested the system. The fix-and-test cycle continued until it resulted in a system that is ready for deployment. In addition to the system's functionality testing, the developers revisited the source codes to ensure that they were organized as planned and are well-documented.

After the system testing, the proponents deployed the SIS on the Web for further testing. They encountered some connectivity errors but were immediately fixed. During the developers' final system testing, they used the Product Quality Model in the ISO 25010:2011 Systems and Software Engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and Software Quality Models to ensure that the developed system is at par with international standards. The product quality model categorizes product quality properties into eight characteristics - functional suitability, reliability, performance efficiency, usability, security, compatibility, maintainability, and portability ("ISO/IEC 25010:2011 (En)", n.d.). Each characteristic has a set of related sub-characteristics. The developers did not evaluate the system's usability because the actual users should assess this aspect. Table 2 shows the results of the evaluation.

Table 2

Results of system evaluation using ISO 25010: 2011

Software / System Characteristics	%	Compliance
Functional Suitability	100%	Full Compliance
Performance Efficiency	100%	Full Compliance
Compatibility	100%	Full Compliance
Reliability	100%	Full Compliance
Security	100%	Full Compliance
Maintainability	100%	Full Compliance
Portability	100%	Full Compliance

Since the software possesses all the sub-characteristics, there is full compliance in all the main characteristics. This means that the AMPMNHS Student Information System is functionally suitable, performs efficiently, compatible with the software that it interacts with, and hardware that it utilizes, reliable, secure, maintainable, and portable.

The final features of the developed system are the following:

- Only authorized users can access the system.
- Management of Student Personal Information, Employee Personal Information, Curriculums, Subjects, Subjects included in a curriculum, Section, Student Subjects, School Year, Grade Level, Student Grades.
- Generation of reports such as Form 137, Student Information Sheet, other reports. Reports may be directly fed to the printer or exported with the following file formats pdf, word processing, electronic spreadsheet.

Management of records includes the following processes: add a new record, search a record or records with filter options, edit a record (authorized users can edit only editable data), delete a record (an authorized user can delete only a record that has not been used yet in any transaction), and archive a record.

Conclusion and Recommendations

The Student Information System of Alejo M. Pacalso Memorial National High School was developed to manage Junior and Senior High School curriculums and subjects and student-related data such as personal information, section, subjects, and grades, and generate relevant reports. It was tested thoroughly by the developers using different testing techniques. The final testing showed that it satisfied the users' requirements. Also, it is fully compliant with the characteristics of software based on ISO 25010:2011. The system is ready for use by AMPMNHS. However, due to current restrictions, the system is yet to be deployed in the school's environment. Deployment can be done when restrictions ease up and the risks to the developers and school representatives are lowered.

The system may be enhanced to integrate students' billing and payments, discipline-related data, guidance and counseling records, student performances, and other relevant information about students and a more efficient backup and restore processes. Having comprehensive information about a particular student or a group of students supports data-driven decision-making, research, and planning by the administration and the teachers.

After using the system for at least one (1) school year, a follow-on research may be conducted to determine the system's extent of effectiveness in terms of service and productivity improvements.

References

Abugabah, A., Sanzogni, L. & Poropat, A. (2009). *The impact of information systems on user performance: A critical review and theoretical model.* CORE, WASET, 2009, core.ac.uk/reader/143854356.

Bourque, P. & Fairley, R.E. (Eds.). (2014*). Guide to the Software Engineering Body of Knowledge, Version 3.0.* https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3

DepEd (n.d.). *DO 26, S. 2015* – Learner Information System (LIS) and Enhanced Basic Education Information System (EBEIS) Updating for Beginning of School Year (BOSY) 2015-2016. http://www.deped.gov.ph/2015/06/26/do-26-s-2015-learner-information-system-lis-and-enhanced-basic-education-information-system-ebeis-updating-for-beginning-of-school-year-bosy-2015-2016/

Dino, L. (2017). *State of Basic Education [Powerpoint slides].* https://peac.org.ph/wp-content/uploads/2017/12/DINO-State-of-Basic-Education_Usec-Dino-Nov-27-2017-Copy-ilovepdf-compressed.pdf

Harrington, J. L. (2009). *Relational database design and implementation clearly explained.* Morgan Kaufmann/Elsevier.

ISO/IEC 25010:2011(2011). *ISO, Online Browsing Platform (OBP)*. www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en.

IBM Cloud Education (2020, October 28). *Three-tier Architecture.* Retrieved from https://www.ibm.com/cloud/learn/three-tier-architecture

Link, J. (2004). *Unit Testing in Java How Tests Drive the Code.* https://www.sciencedirect.com/book/9781558608689/unit-testing-in-java?via=ihub=

Microsoft (2020). *Description of the database normalization basics*. https://docs.microsoft.com/en-us/office/troubleshoot/access/database-normalization-description.

O'Brien, J. A. & Marakas, G. M. (2011). *Management information systems, 10th ed.* New York: McGraw-Hill/Irwin.

Oracle (n.d.). *Application Server TopLink Application Developer's Guide.* https://docs.oracle.com/cd/B14099_19/web.1012/b15901/archtrs004.htm

Shah, M. (2014). Impact of Management Information Systems (MIS) on School Administration: What the Literature Says. *Procedia - Social and Behavioral Sciences, 116*, 2799-2804. doi:10.1016/j.sbspro.2014.01.659

Sharma, A., Kumar, M., & Agarwal, S. (2015). A complete survey on software architectural styles and patterns. P*rocedia Computer Science, 70*, 16-28. doi:10.1016/j.procs.2015.10.019